

**CLAIMS**

1. A method for executing a network-based distributed application, the method comprising:

executing application instances of the distributed application in application containers;

5 calculating quality of service metrics for each application instance; and

distributing application workload among the application instances using a decentralized workload management layer based on the quality of service metrics.

2. The method of claim 1, further comprising associating application containers with autonomous workload management elements, the workload management elements forming the workload management layer.

3. The method of claim 2, further comprising coordinating the application instances through a coordination mechanism coupled to the workload management layer.

4. The method of claim 1, wherein distributing application workload among the application instances further comprises reducing workload assigned to an application container when the quality of service metrics reach an overload threshold value.

5. The method of claim 4, wherein reducing workload assigned to the application container further comprises:

examining an encoding of work unit groups provided by each application instance;

5 splitting a currently assigned work unit group into smaller work unit groups;

assigning at least one of the smaller work unit groups to other application containers; and

10 utilizing a coordination mechanism to update changes in workload assignments to the other application containers

6. The method of claim 1, wherein distributing application workload among the application instances further comprises increasing workload assigned to the application container when the quality of service metrics reach an under-load threshold value.

7. The method of claim 6, wherein increasing workload assigned to the application container further comprises:

examining an encoding of work unit groups provided by each application instance;

5 combining at least two currently assigned work unit groups into a smaller work unit group;

assigning the smaller work unit group to the application container; and

10 utilizing a coordination mechanism to update changes in workload assignments to the other application containers.

8. The method of claim 1, further comprising dividing workload assigned to a single application instance to at least two application instances if a quality of service metric reaches an overload threshold.

9. The method of claim 1, further comprising:

dividing a total workload performed by the distributed application among the application instances;

5 assigning each of the application instances a fractional workload; and

filtering client requests at the application containers based on the fractional workload assigned to the application instances.

10. The method of claim 9, further comprising migrating a client from a first application container to a second application container if workload from the client is not assigned to the application instance executing at the first application container.

11. The method of claim 10, further comprising labeling client requests such that application containers can determine if the requests belong to the fractional workload assigned to the application instances.

12. The method of claim 1, further comprising receiving the application instances from application loaders.

13. A system for executing a distributed computer application, the system comprising:

one or more application containers configured to execute an application instance of the distributed application and determine  
5 quality of service metrics for the application instance; and

one or more workload management elements forming a decentralized workload management layer, each workload management element is configured to be associated to one of the application containers and to assign a workload to the application container based on the quality of  
10 service metrics received by the application container.

14. The system of claim 13, wherein each workload management element is further configured to autonomously increase and decrease the assigned workload to its associated application container.

15. The system of claim 14, wherein each workload management element is further configured to divide the assigned workload to two or more application containers if the assigned workload to its associated application container is to be decreased.

16. The system of claim 14, wherein each workload management element is further configured to combine the assigned workload of two or more application containers if the assigned workload to its associated application container is to be increased.

17. The system of claim 13, wherein each application container is further configured to pass inbound packets to executing application instances when the inbound packets belong to the its assigned workload, and to pass inbound packets to its associated workload management  
5 element when the inbound packets do not belong to its assigned workload.

18. The system of claim 13, further comprising workload tags coupled to data packets of application containers, the workload tags

configured to allow application containers to identify whether the inbound packets belong to its assigned workload.

19. The system of claim 13, further comprising a coordination mechanism configured to workload management elements to locate each other and determine the current work assignments of each application container.

20. The system of claim 13, further comprising an application loader configured to provide executable application code to application containers.

21. A computer program product embodied in a tangible media comprising:

computer readable program codes coupled to the tangible media for executing a network-based distributed application, the computer

5 readable program codes configured to cause the program to:

execute application instances of the distributed application in application containers;

receive quality of service metrics for each application instance; and

10 distribute application workload among the application instances using a decentralized workload management layer based on the quality of service metrics.

22. The computer program product of claim 21, further comprising program code configured to associate application containers with workload management elements, the workload management elements forming the workload management layer.

23. The computer program product of claim 22, further comprising program code configured to coordinate the application instances through a coordination mechanism coupled to the workload management layer.

24. The computer program product of claim 21, wherein the program code configured to cause the program to distribute application

workload among the application instances further comprises program code  
to the program to reduce workload assigned to an application container  
5 when the quality of service metrics reach an overload threshold value.

25. The computer program product of claim 23, wherein the  
program code configured to cause the program to reduce workload  
assigned to the application container further comprises program code to  
the program to:

5 examine an encoding of work unit groups provided by each  
application instance;  
split a currently assigned work unit group into smaller work unit  
groups;  
assign at least one of the smaller work unit groups to other  
10 application containers; and  
utilize a coordination mechanism to update changes in workload  
assignments to the other application containers.

26. The computer program product of claim 21, wherein the  
program code configured to cause the program to distribute application  
workload among the application instances further comprises program code  
to the program to increase workload assigned to the application  
5 container when the quality of service metrics reach an under-load  
threshold value.

27. The computer program product of claim 26, wherein the  
program code configured to cause the program to increase workload  
assigned to the application container further comprises program code to  
the program to:

5 examine an encoding of work unit groups provided by each  
application instance;  
combine at least two currently assigned work unit groups into a  
smaller work unit group;  
assign the smaller work unit group to the application container;  
10 and  
utilize a coordination mechanism to update changes in workload

assignments to the other application containers.

28. The computer program product of claim 21, further comprising program code configured to divide workload assigned to a single application instance to at least two application instances if a quality of service metric reaches an overload threshold.

29. The computer program product of claim 21, further comprising program code configured to:

divide a total workload performed by the distributed application among the application instances;

5 assign each of the application instances a fractional workload;  
and

filter client requests at the application containers based on the fractional workload assigned to the application instances.

30. The computer program product of claim 29, further comprising program code configured to migrate a client from a first application container to a second application container if workload from the client is not assigned to the application instance executing  
5 at the first application container.

31. The computer program product of claim 30, further comprising program code configured to label client requests such that application containers can determine if the requests belong to the fractional workload assigned to the application instances.

32. The computer program product of claim 21, further comprising program code configured to receive the application instances from application loaders.